ELSEVIER

Computational Materials Science 4 (1995) 103-116

COMPUTATIONAL MATERIALS SCIENCE

Numerical methods for computing interfacial mean curvature

J.W. Bullard [a], E.J. Garboczi [a], W.C. Carter [b], E.R. Fuller Jr. [b]

[a] *Building Materials Division, National Institute of Standards and Technology, Gaithersburg, MD 20899-0001, USA*

[b] *Ceramics Division, National Institute of Standards and Technology, Gaithersburg, MD 20899-0001, USA*

Received 1 September 1994; accepted 2 November 1994

Abstract

A procedure is described for computing the mean curvature along condensed phase interfaces in two or three dimensions, without knowledge of the spatial derivatives of the interface. For any point P on the interface, the method consists of computing the portion of volume enclosed by a small template sphere, centered on P, that lies on one side of the interface. That portion of the template volume is shown to be linear in the mean curvature of the surface, relative to the phase lying on the opposite side of the interface, to within terms that can usually be made negligible. An analogous procedure is described in two dimensions. Application of the procedure to compute the mean curvature along a digitized surface is demonstrated. A burning algorithm can be included to improve computational accuracy for interfaces having sharp curvature fluctuations. A minor extension of the method allows computation of the orientation of an interfacial element relative to a fixed reference frame.

1. Introduction

Recent developments in the use of computer models that operate on digital images of material microstructures have made possible a quantitative examination of the influence of the details of geometrically complex microstructures on macroscopic properties [1–4]. One class of processes that these models can potentially simulate is sintering phenomena [3,4]. Since sintering and related processes are often driven primarily by local differences in curvature throughout the microstructure, sintering models that operate on digitized microstructure images require a numerical method for computing curvature along digitized surfaces. In this paper, we describe and demonstrate one such method that has been used successfully in recent sintering simulations [3,4], putting it on a firm theoretical footing in both 3D and 2D. Analytical and numerical errors inherent in the method are explicitly stated and assessed.

1.1. Background

The mean curvature of an infinitesimal element along a condensed-phase interface represents the quantity $\delta A/\delta V$, where δA is the incremental change in the element's area when it is normally displaced by local addition of material of volume δV [5]. Since a finite positive energy always accompanies formation of a unit area of an interface, mean curvature plays an important role in governing the thermodynamics of interfacial phenomena.

The effects of mean interfacial curvature are manifested in a wide range of both equilibrium and non-equilibrium phenomena. An example of the former is the dependence of the equilibrium vapor pressure of a one-component liquid drop on the drop size. If there are no surface tractions or body forces deforming the drop, its equilibrium shape is a sphere of radius R, and

0927-0256/95/$09.50
SSDI 0927-0256(95)00014-3

its vapor pressure, p, is related to its mean curvature [1], $H = R^{-1}$, by the Kelvin equation,

$$\ln\frac{p}{p_o} = \frac{2\Omega\gamma H}{k_B T}, \qquad (1)$$

where p_o is the equilibrium vapor pressure above a planar surface, Ω is the molecular volume of the species composing the drop, γ is the surface free energy density, k_B is the Boltzmann constant, and T is the absolute temperature. Non-equilibrium phenomena in which mean curvature plays an important role include those processes acting to reduce the overall interfacial energies during microstructural development of polycrystalline solids: sintering [6,7], grain growth [8], and Ostwald ripening [9]. For example, the driving force for diffusive mass transport during these processes is the gradient in chemical potential, μ, between portions of the interfaces (assuming that no other driving forces, like those for phase transformations, are present). When the interfacial energy density, γ, is independent of crystallographic orientation, then the excess contribution to μ along the interface is determined by H. The dependence on mean curvature of $\nabla_s\mu$ along the surface can then be found from the Gibbs-Thomson equation [10] for an interface composed of one chemical species,

$$\nabla_s\mu = 2\gamma\Omega\nabla H. \qquad (2)$$

Eq. (2) indicates why theoretical calculations of curvature-driven processes are usually difficult. An analytic description of H, as a function of position along the interface, is required in order to calculate the instantaneous driving force for curvature-driven mass transport. When the position of a 2D surface can be described by an analytic function, an analytical calculation of H is tractable at any twice-differentiable point on the surface. If ∇H is also defined at that point, the driving force for mass transport is accessible from Eq. (2), provided that the surface energy is isotropic. Unfortunately, the surfaces and interfaces in random, porous polycrystalline solids are tortuous and usually defy analytical description. Furthermore, curvature-driven transport generally causes the surface shape to evolve with time. Modeling such transport therefore demands tracking the evolving surface, a generally formidable task. For these reasons, theoretical studies of mass transport during sintering have been limited to simple geometries, like the two-sphere model of Kingery and Berg [11], and more recently the linear particle array models of Carter and Cannon [12] and Kellett and Lange [13]. These and other models have proven to be very important in elucidating the major features of curvature-driven processes. However, further progress in modeling the development of real microstructures requires a different approach for assessing and tracking mean curvature variations along geometrically complex interfaces.

One way of approximating the curvature of an arc or an interface of arbitrary shape is to first represent the interface as a polynomial fit to consecutive points along the interface, and to then compute the necessary spatial derivatives [14]. A related method consists of constructing two orthogonal osculating circles by fitting each circle to three consecutive points along the surface coplanar with that circle [15]. The curvature is then determined by the sum of the inverses of the radii of the osculating circles. These methods require, at any point P on the surface, selection of several closely-spaced points within a small neighborhood of P to which a curve is fit. The shape of that curve, and therefore the computed curvature at P, can therefore depend heavily on the choice of fitting points, especially at regions of high curvature.

A third procedure for obtaining relative estimates of curvature consists, at any point P on the interface, of computing the portion of the volume enclosed by a small template sphere, centered at P, that lies on one side of the interface. This computation is particularly straightforward when the surface can be adequately represented by a collection of discrete elements or pixels. A similar technique has been employed to obtain qualitative curvature distributions in simulations of diffusion-controlled growth of aggregates in 2D with finite values of γ [16-18] and for cellular automata simulations of curvature-driven sintering in 2D [4] and 3D [3]. For those simulations, the investigators obtained a relative curvature measure by counting the number of pixels external to the surface but within

[1] Throughout this paper we adopt the convention from differential geometry of denoting H at any point P as the arithmetic mean of the two principal curvatures, κ_1 and κ_2, where κ_1 and κ_2 are the reciprocals of the radii of two mutually orthogonal "osculating" circles constructed tangent to the surface at P. We also adopt the convention of sintering theory of taking a principal curvature to be positive if the surface normal vector points away from the center of curvature.

either a square [4,16–18] or a spherical [3] template centered on a given surface pixel. Such template schemes represent a generalization of the method, used by Holm and coworkers for Potts-model-type Monte Carlo simulations of grain growth [19] on a discrete lattice, of computing the spin Hamiltonian by counting the number and type of dissimilar nearest neighbors at a given grain boundary site. Although phrased in terms of a lattice Hamiltonian, their model qualitatively predicts the same grain boundary motion as that predicted by assuming curvature-driven motion [19].

Previous use of the template method for computing curvature has been justified by the demonstration that the pixel count described in the preceding paragraph is roughly proportional to mean curvature [4,16–18] for a square template[2]. In 3D, Bentz and coworkers [3] showed analytically that, for the special case of a spherical template centered on the surface of a much larger sphere, the enclosed volume is exactly proportional to the mean curvature of the larger sphere. Analytical calculations were not provided for non-spherical surfaces [3].

In this paper we present a more general justification of the validity of the template method for computing curvature than has been given in previous papers [3,4,16–18] by analytically deriving the relationship between mean curvature and the spherically-bounded volume described in the preceding paragraph, for an arbitrary curve (in 2D) or surface (in 3D). The derivation demonstrates that this bounded volume is approximately linear in the mean curvature, to within higher-order correction terms. Estimates of the magnitudes of the correction terms for typical values of the curvature show these terms to be negligible in most cases. We then describe and illustrate application of the procedure to computing the discrete analogue of mean curvature along interfaces whose shapes are represented as a collection of discrete pixel elements. Digitization of the surface introduces larger errors that can cause the result of an individual curvature computation to deviate substantially from the true mean curvature. Even so, the method produces a reliable measure of the spatial distribution of curvature along non-uniform interfaces and throughout complex microstructures. This latter quality of the method makes it a useful tool for sintering simulations.

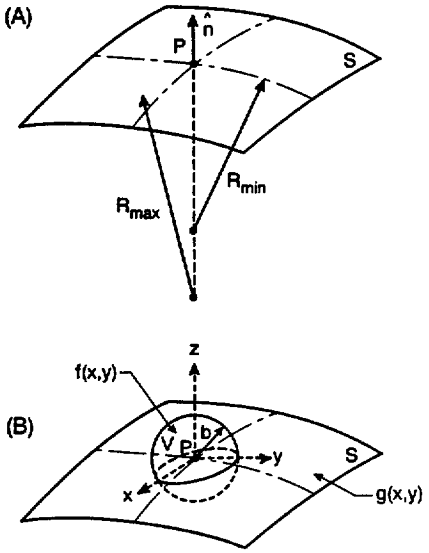

Fig. 1. (A) Point P centered within a two-dimensional surface element S, characterized by the two principle radii of curvature R_{min} and R_{max}. (B) Template sphere of radius b centered on P. The Cartesian reference frame is oriented such that the z-axis is normal to S at P, and such that P is at the origin. V (see text) is the portion of the template volume that lies entirely to one side of S (in this figure, the upper side).

2. Analysis

We wish to analytically justify the assertion that the mean curvature of a sufficiently small surface element of a condensed phase α is linear in (within an additive correction term) that portion of the volume of a small sphere, centered on the surface element, that does not contain α. The meaning of the terms "sufficiently small surface element" and "small sphere" will become apparent during the analysis.

Consider a point P lying within a small element S of a surface that bounds phase α, with outward unit normal vector (pointing away from α), as in Fig. 1(A). Assume that S is specified by a function, g, of two spatial coordinates, and that g is continuous on the element to at least second order in those coordinates. The principal radii of curvature at P, $R_{min} = \kappa_{max}^{-1}$ and $R_{max} = \kappa_{min}^{-1}$ ($R_{min} < R_{max}$), can in general have

[2] For square templates in 2D and cubic templates in 3D, the curvature computation is subject to variability along surfaces due to anisotropy in the template shape (see Ref. [4]). This effect is significantly reduced by using circular (2D) or spherical (3D) templates.

any sign. Next, consider a small imaginary "template" sphere of radius b ($b << R_{min}$) centered at P (shown in Fig. 1(B)). Denote by V that portion of the sphere volume that does not intersect α.

Without loss of generality, we may orient a Cartesian coordinate frame such that (i) the tangent plane to S, passing through P is the xy-plane with P at the origin, and (ii) the projection of the two principal lines of curvature at P onto the xy-plane are the x and y axes (Fig. 1(B)). With this choice of coordinates, the slope of the tangent plane at P is zero, and we assume that S can be represented as a convergent Taylor expansion of $g(x,y)$ about the origin:

$$g(x,y) = \frac{1}{2}\left[\left(\frac{\partial^2 g}{\partial x^2}\right)_{x=y=0} x^2 + \left(\frac{\partial^2 g}{\partial y^2}\right)_{x=y=0} y^2 + 2\left(\frac{\partial^2 g}{\partial x \partial y}\right)_{x=y=0} xy\right] + O(g_j''' x^j y^{3-j}) \quad (3)$$

where

$$g_j''' = \left(\frac{\partial^3 g}{\partial x^j \partial y^{3-j}}\right)_{x=y=0}.$$

Because of the placement of the coordinate system, the principal curvatures κ_{max} and κ_{min} at P are equal to the negative of the first and second partial derivatives, respectively, appearing in Eq. (3). Furthermore, by comparing Eq. (3) with an analogous expansion using orthogonal curvilinear parameters that define lines of curvature, it can be shown that the third 2nd-order partial derivative in Eq. (3) is zero [20,21]. Finally, we rewrite Eq. (3) in terms of dimensionless variables that scale with the radius of the template sphere, b ($X = x/b$, $K_{max} = b\kappa_{max}$, etc.),

$$Z_s = \frac{z_s}{b} = G(X,Y) = -\frac{1}{2}(K_{max}X^2 + K_{min}Y^2) + O(\epsilon_j X^j Y^{3-j}), \quad (4)$$

where

$$\epsilon_j = \left(\frac{\partial^3 G}{\partial X^j \partial Y^{3-j}}\right)_{X=Y=0} \qquad (j = 0,1,2,3).$$

We assume throughout this analysis that the maximum magnitude of the four ϵ_j terms, ϵ, is small compared to unity. In other words, we restrict the analysis to surfaces for which the expansion Eq. (4) is a good approximation.

It will prove convenient to use cylindrical coordinates R ($= r/b$), θ, and z rather than Cartesian coordinates. Then Eq. (4) can be rewritten as

$$Z_s = -\frac{1}{q}R^2 + O(\epsilon), \quad (5)$$

where $q = 2(K_{max}\cos^2\theta + K_{min}\sin^2\theta)^{-1}$. The scaled volume $\nu = Vb^{-3}$ is then given by the integral expression

$$\nu = \frac{4}{3}\pi - \int_0^{2\pi} \mathrm{d}\theta \int_0^{R^*(\theta)} (Z_s - Z_t) R \,\mathrm{d}R, \quad (6)$$

where Z_t is the equation for the lower half of the template sphere,

$$Z_t = -\sqrt{1 - R^2}, \quad (7)$$

and $R^*(\theta)$ is determined by $Z_s = Z_t$, and can easily be shown to be

$$R^*(\theta) = \sqrt{\frac{\Gamma}{\beta} - \frac{\Gamma^2,}{q^2\beta^3}}, \quad (8)$$

where

$$\Gamma = 1 - O(\epsilon^2) \approx 1, \quad \beta = 1 - \frac{2O(\epsilon)}{q} \approx 1 - KO(\epsilon)$$

to order ϵ, with $K = (K_{max} + K_{min})$. Substituting Eqs. (4) and (7) into Eq. (5) gives

$$\nu = \frac{4}{3}\pi + \int_0^{2\pi} \mathrm{d}\theta \left[\frac{1}{4q}R^4 - \frac{1}{2}R^2 O(\epsilon) + \frac{1}{3}(1 - R^2)^{3/2}\right]_0^{R^*(\theta)},$$

and substituting Eq. (8) gives

$$\nu = \left(\frac{2}{3} - \frac{O(\epsilon)}{\beta}\right)\pi + \frac{1}{4\beta^2}\int_0^{2\pi} \frac{\mathrm{d}\theta}{q} + \frac{O(\epsilon)}{2\beta^3}\int_0^{2\pi} \frac{\mathrm{d}\theta}{q^2} - \frac{1}{2\beta^4}\int_0^{2\pi} \frac{d\theta}{q^3} + \frac{1}{4\beta^6}\int_0^{2\pi} \frac{\mathrm{d}\theta}{q^5} + \frac{1}{3}\int_0^{2\pi} \left(1 - \frac{1}{\beta} + \frac{1}{q^2\beta^3}\right)^{3/2} \mathrm{d}\theta.$$

Using the definition of q produces

$$\nu = \left(\frac{2}{3} - \frac{O(\epsilon)}{\beta}\right)\pi + \frac{\pi}{8\beta^2}(K_{max} + K_{min}) - \frac{\pi}{128\beta^4}\Theta(K^3) + \frac{1}{3}\int_0^{2\pi}\left(1 - \frac{1}{\beta} + \frac{1}{q^2\beta^3}\right)^{3/2} \mathrm{d}\theta, \tag{9}$$

where

$$\Theta(K^3) = (K_{max} + K_{min})^3 + 4(K_{max}^3 + K_{min}^3).$$

If $\beta < 1$, then we may expand β^{-n} according to

$$\beta^{-n} = 1 + nKO(\epsilon)$$

to order ϵ. Substituting into Eq. (9) yields

$$\nu = \frac{2}{3}\pi + \frac{\pi}{8}(K_{max} + K_{min}) - \frac{\pi}{384}\Theta(K^3) - \pi O(\epsilon)\left[1 - \frac{1}{8}K(K_{max} + K_{min}) - \frac{1}{8}\Lambda(K^2) - \frac{9}{128}K\Theta(K^3)\right]. \tag{10}$$

where

$$\Lambda(K^2) = (K_{max} + K_{min})^2 + 2(K_{max}^2 + K_{min}^2),$$

From Eq. (10)

$$\begin{aligned} H &= \frac{1}{2b}(K_{max} + K_{min}) = \frac{4}{\pi}\frac{\nu}{b} - \frac{8}{3b} - \frac{1}{b}\left[\frac{1}{96}\Theta(K^3) + O(\epsilon)(\Xi)\right] \\ &= \frac{4}{\pi}\frac{V}{b^4} - \frac{8}{3b} - \frac{1}{b}\left[\frac{1}{96}\Theta(K^3) + O(\epsilon)(\Xi)\right] \end{aligned} \tag{11}$$

where the symbol Ξ represents the last term in brackets in Eq. (10). The mean curvature H is therefore approximately linear in V, provided that $K = \kappa b << 1$ (which will always be true for a given application if the template radius b is chosen to be small compared to the inverse of the maximum expected curvature). The error in the linear approximation, the bracketed term in Eq. (11), contains contributions from two sources: 1) the error in the 2nd-order approximation of the surface, and 2) higher order curvature terms. The error is usually small compared to the first two terms on the right hand side of Eq. (11), certainly when $K_{max} <<$ 1, but even when this limitation is not obeyed. For example, choosing values of $K_{min} = 0.25$, $K_{max} = 0.5$, and $b = 0.5$, which do not obey $K_{max} << 1$, then the first term in brackets in Eq. (11) represents only a 2.7% accuracy error (the magnitude of the second term in brackets in Eq. (11) of course depends on the accuracy of Eq. (4) in approximating the surface).

The second term on the right side of Eq. (11) also indicates one major effect of the template radius, b: The maximum magnitude of curvature that can be measured using the linear approximation is inversely proportional to b. Smaller templates therefore increase the *range* of measurable curvatures in the continuum limit. We will see in the next section, though, that when applying the method to digitized representations of surfaces, smaller templates decrease the *resolution* of the curvature measurement.

A different form for $R^*(\theta)$, Eq. (8), will generally result for different surface shapes. For example, a surface element that is a portion of a sphere gives

$$R^*(\theta) = \sqrt{1 - \frac{1}{4q^2}}. \tag{8a}$$

From Eq. (8a) it is easily shown that

$$H = \frac{4}{\pi}\frac{\nu}{b} - \frac{8}{3b}. \tag{11a}$$

In other words, the linear approximation is exact for spherical surface elements.

To summarize, the linear approximation between mean curvature and V, Eq. (11), provides estimates of the mean curvature, typically accurate to within ± 10%, at any point on a surface that can locally be approximated by a 2nd-order expansion, Eq. (4). The error in Eq. (4) generally increases with increasing displacement from the origin, that is, Eq. (4) becomes less accurate for larger template radii. Finally, it should be mentioned that the analysis is not strictly appropriate at points at which there is a discontinuity in the first or second order spatial derivatives, such as at sharp corners and edges. At such points the magnitude of the true mean curvature is unbounded, while the volume calculations shown in this section establish an upper limit on the magnitude at any point that is dictated by the volume of the template. Application of this method at these points can still, however,

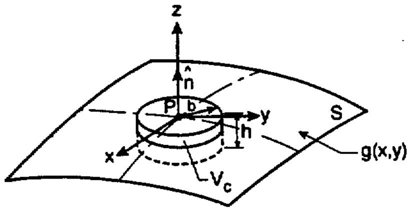

Fig. 2. Right circular cylinder template positioned such that one of its bases of radius b is tangent to the surface element S at P. The template height h is great enough to ensure that the other base of the cylinder does not intersect S. The Cartesian frame is oriented as in Fig. 1. V_c (see text) is the portion of the template volume that lies entirely to one side of S (in this figure, the upper side).

give some idea of an "apparent" curvature because the computed magnitude of the curvature will depend on the difference in slopes on either side of the singularity (razor edges, for example, will yield a higher computed value of curvature than right edges).

We now proceed to describe a few simple extensions of the template concept that are potentially useful under certain circumstances.

2.1. Cylindrical template

We can equally well use a right-circular-cylinder template of radius b and height h instead of the spherical template of radius b used in the preceding analysis. In this case, we position the center of one base of the cylinder on the point P so that the cylinder axis coincides with the surface normal vector $\hat{\mathbf{n}}$, and then require h to be sufficiently large that the other base lies entirely beneath the surface described by Eq. (4) (see Fig. 2). The portion of the scaled cylinder volume ν_c lying above the surface is then easily calculated, using Cartesian coordinates, by

$$\nu_c = -\int_{-1}^{1} \mathrm{d}X \int_{-\sqrt{1-X^2}}^{\sqrt{1-X^2}} G(X,Y)\,\mathrm{d}Y, \qquad (12)$$

where the origin of the Cartesian frame again coincides with the point P in question, $G(X,Y)$ is given by Eq. (4), and the z-axis coincides with the cylinder axis. Integrating Eq. (12) gives

$$\nu_c = \frac{\pi}{8}(K_{max} + K_{min}) + O(\epsilon), \qquad (13)$$

or equivalently,

$$H = \frac{4}{\pi b}[\nu_c - O(\epsilon)]. \qquad (14)$$

Therefore, the only error incurred in computing the mean curvature by this method is due to any error in the 2nd-order approximation of the surface, $O(\epsilon)$, in Eq. (4). Implementation of this method all along an interface, however, is somewhat more complicated than that of the sphere method, since the cylinder method requires knowledge of the surface normal direction before the cylinder can be correctly placed. We therefore will restrict attention to the spherical template when discussing computer implementation in three dimensions.

2.2. Two-dimensional calculations: circular template

The curvature of an arc bounding a 2D "membrane" of phase α can be computed by using a circular template in the same manner as the spherical template is used in three dimensions. Any infinitesimal arc element possessing curvature κ at the origin of a polar coordinate system can be represented as a 2nd-order Taylor expansion about the origin. Proceeding as we did to arrive at Eq. (5) in 3D,

$$\theta(R) = \sin^{-1}\left[-\frac{1}{2}KR + \frac{1}{8}(KR)^3 + O(\epsilon)\left(\frac{1}{2} - \frac{1}{4}(KR)^2\right)\right], \qquad (15)$$

where θ is the angle from the tangent line passing through P, $K = \kappa b$, and ϵ is the 2D analogue to ϵ appearing in Eq. (5). The portion of scaled area, $a = Ab^{-2}$, of the template circle that excludes phase α is then given by

$$a = \int_0^1 R\,dR \int_{\theta(R)}^{\pi-\theta(R)} \mathrm{d}\theta \qquad (16)$$

which integrates to give, after expanding $\theta(R)$ to 3rd order in K,

$$a = \frac{\pi}{2} + \frac{1}{3}K + O(\epsilon)\left(\frac{3}{32}K^2 - \frac{1}{2}\right) \qquad (17)$$

or, equivalently

$$K = 3a - \frac{3\pi}{2} + \left[\frac{1}{8}K^3 - O(\epsilon)\left(\frac{9}{32}K^2 - \frac{3}{2}\right)\right]. \qquad (18)$$

Again, the error in the linear approximation, the term enclosed in square brackets, is usually small compared to the true curvature. For example, taking the surface as a circle for which $K = 0.5$, the error is 1.2%.

As a further generalization of the method in 2D (omitted from the 3D description due to mathematical complexity), one can bias the area elements over which integration is performed by including a dimensionless weighting function of the form

$$W(R) = \frac{w}{R^{n+1}}, \tag{19}$$

where w is a positive real constant and n an integer. A weighting factor like Eq. (19) can potentially be useful because it reduces the dependence of the area integral, Eq. (16), on portions of the surface that are further removed from P. By biasing the computation to that part of the surface very near P, weighting the integral can increase accuracy. Using Eq. (19), Eq. (16) then becomes

$$\begin{aligned} a_w &= \int_0^1 WR\,dR \int_{\theta(R)}^{\pi-\theta(R)} d\theta, \\ &= w \int_0^1 R^{-n}\,dR \int_{\theta(R)}^{\pi-\theta(R)} d\theta. \end{aligned} \tag{20}$$

Examination of Eq. (20) reveals that $n \leq 0$ for the integral to remain bounded. Of course, for $n = -1$ and $w = 1$, Eq. (20) reduces to Eq. (16). Increasingly negative values of n will bias a_w away from P. Since curvature depends on highly localized variations in the surface at P, we will perform the integration for $n = 0$. In that case,

$$K = \frac{2}{w} a_w - 2\pi + \left[\frac{5}{48} K^3 - O(\epsilon) \left(\frac{1}{4} K^2 - 2 \right) \right]. \tag{21}$$

Comparison of this result to Eq. (18) shows that the error in the linear approximation due to higher-order curvature terms can be reduced by almost 20% if $1/R$ weighting is incorporated, but $1/R$ weighting is ineffective for reducing the error caused by approximating the surface to second order. In the remainder of this paper, we will focus on the unweighted result, Eq. (18), since it is easier to implement computationally, and the error term in Eq. (18) is usually small enough already.

3. Application to discretized interfaces

In this section we demonstrate the use of the curvature computation on images of surfaces that have been mapped onto a regular 2D (3D) grid of square (cubic) pixel elements. Each pixel in the grid can be assigned a phase corresponding to those in the image, and any pixel containing phase α, for example, is then an α-β surface pixel if it is in contact along an edge (face) with a β-pixel. The area (volume) calculations described in the previous section therefore reduce to counting pixels within a digitized circular (spherical) template. Discrete mapping of an interface therefore allows a simple computation of mean curvature for every interface element in the grid.

As with any other method that uses digital images, this procedure can be memory intensive, especially when applied in sintering simulations of porous powder compacts. Accurate curvature computation on a single particle surface usually requires that the particle contain at least ~100 pixels in 2D, or ~1000 pixels in 3D. Simulating the sintering of a portion of a compact containing, say, 1000 particles therefore requires a lattice composed of 2×10^6 pixels, or about 1 Mb of memory. In addition, the curvature computation method itself requires some memory allocation, since the position and pixel count of each interfacial pixel must be stored. But if the position and pixel count each use 4 bytes of memory for each interfacial pixel, and if the number of interfacial pixels is about 5% of the total number of pixels (typical for a powder compact with ~1 μm-diameter particles), then the curvature method requires only around 10% extra memory allocation. Typical workstations can accommodate this memory requirement, although much larger-scale simulations may require the memory capacity of a supercomputer.

3.1. Sources of error

When performing the curvature computation in the digital mode, several sources of error are encountered in addition to those described in the previous section. One source of error lies in the digitized approximation

Table 1
Deviation in area of a circle composed of discrete square pixels from the area of a true circle of the same diameter

Circle diameter (ξ)	% Area Error
5	7.0 %
9	8.5 %
15	0.2 %
21	0.8 %
31	- 0.8 %
41	- 0.5 %

Table 2
Deviation in volume of a sphere composed of discrete cubic pixels from the volume of a true sphere of the same diameter

Sphere diameter (ξ)	% Volume Error
5	23.8 %
9	1.9 %
15	1.3 %
21	2.0 %
31	- 0.5 %
41	0.1 %

of the template. As Table 1 shows, the area of a "circular" template depends significantly on the number of pixels used to construct it. Assigning length unit ξ to each pixel edge, the area within the template is not equal to the true area of a circle of the same diameter. But the error in template area becomes quite small when a sufficient number of pixels are used, and Table 1 shows that a 15-ξ or greater diameter gives an excellent approximation to the true circle area (Table 2 shows similar information for the volume of spherical templates). Table 1 also shows that using more pixels to refine the circle shape, by a factor of 2 or 3 beyond a 15-ξ diameter, does not significantly improve the area approximation, while Table 2 seems to indicate that increasing a sphere's diameter over this same range does improve the approximation to a true sphere's volume (although the error must approach zero in the limit of infinite diameter). Employing larger templates also increases the portion of surface included in the curvature computation, and for this reason generally causes additional error due to possible non-uniformity of curvature over larger areas. Furthermore, the CPU time required for each curvature computation will increase roughly as the square of the template diameter in 2D, and as the diameter cubed in 3D. Relatively small templates are therefore particularly advantageous in terms of computational accuracy and speed, as long as the area/volume accuracy of the template is adequate.

Another source of error when using digitized templates is the finite resolution of the computation. If the template is composed of N pixels, then only N different values of the curvature are resolvable by this method. Increasing the resolution by increasing N will reduce this problem, but at the same time will tend to magnify the errors described in the previous paragraph. Therefore, for a given application the user must decide the relative importances of accuracy, speed, and resolution when choosing the size of the template.

Further sources of error arise from the discrete approximation of the surface shape as step-like shifts in position, which can alter both the accuracy and precision of the curvature computation. Although a linear relation between curvature and the pixel count is still expected, the slopes and intercepts predicted from Eqs. (11) and (18) are not likely to be correct due to the discrete approximation of the surface and template. When applying the method to a digital image, it is necessary to make an "experimental" determination of the correct values of the slope and intercept in the following way. For a given template diameter, the true mean curvature of a circle (or sphere in 3D) can be plotted against the average value of the pixel counts resulting from each site along the surface. We use averaging because, as will be shown shortly, the pixel-to-pixel variation in computed curvature over a surface with nominally constant non-zero curvature can be substantial, although the average value over the entire surface is quite accurate (see Fig. 4). By repeating for circles (spheres) of varying radii, one can use linear regression to determine the equation of the line that best fits the collection of plotted points. Such a plot is shown for 2D in Fig. 3, using a circular template with diameter $2b = 15\xi$. The predicted continuum equation (now in terms of the physical quantities κ, A, and b),

$$\kappa \approx \frac{3}{b^3}A - \frac{3\pi}{2b} \qquad (18)$$

becomes, in the discrete approximation[3]

[3] We center the template on the center of a surface pixel to achieve this result. Slightly different values for the numerators would result if the template were centered on, for example, a pixel corner.

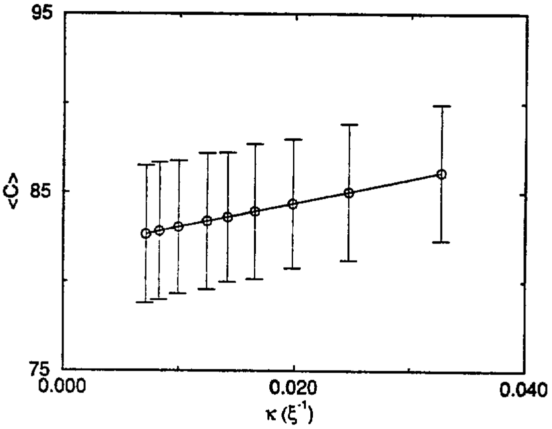

Fig. 3. Pixel count at surface sites along a circle of radius κ^{-1}, plotted against the true curvature of the circle in the continuum limit. Template circle diameter = 15 ξ. Each point is the average of the pixel count computed for all surface sites, and error bars represent ±1 standard deviation.

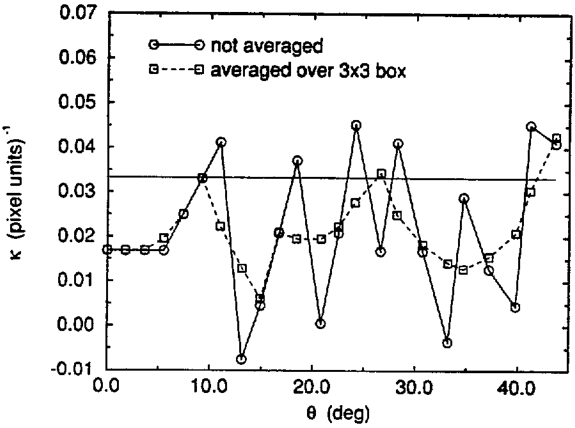

Fig. 4. Computed curvature along one-eighth of the surface of a circle (diameter = 61 ξ), using a template circle (diameter = 15 ξ), showing the effect of averaging the computation over all surface pixels within a 3 × 3 box centered on a given surface pixel P. θ, plotted along the x-axis, is $\tan^{-1}(y/x)$, where x and y are the x- and y-position of P relative to the circle center.

$$\kappa = \frac{2.807}{b^3}\langle C\rangle - \frac{2.586\pi}{2b}, \qquad (22)$$

where $\langle C\rangle$ is the average pixel count. In 3D, using a template with diameter $2b = 9\ \xi$, the predicted continuum equation,

$$H \approx \frac{4}{\pi b^4} - \frac{8}{3b} \qquad (11)$$

becomes

$$H = \frac{3.623}{\pi b^4}\langle C\rangle - \frac{6.337}{3b}. \qquad (23)$$

The diameter of the template sphere modestly affects the values appearing in the numerators of these fitted relations because digitized templates of different diameter have slightly different shapes. Using a template sphere diameter of 15 ξ, linear regression gives instead of Eq. (23),

$$H = \frac{3.868}{\pi b^4}\langle C\rangle - \frac{7.179}{3b}. \qquad (24)$$

The curvature value obtained for an individual surface pixel, using a relation like Eq. (22) or (23), will generally suffer from precision error. This is because, even along surfaces of constant curvature, the discrete approximation of the surface introduces variation in the computed curvature (indicated by the error bars in Fig. 3). For example, Fig. 4 shows the variation in computed curvature with position along the surface of a digitized circle with a 61 ξ-diameter (κ = 0.033 ξ^{-1}) (solid line). For this figure, the template diameter is 15 ξ, and Eq. (22) was used to compute the curvature. The arithmetic mean of the 23 computed values is 0.0332 ξ^{-1}, and therefore the mean has a combined standard uncertainty (u_c) [22], or accuracy, of ≈ 0.0001 ξ^{-1} (1%). But u_c of the individual computations (precision), measured by one standard deviation, is 0.0297 ξ^{-1} (±89% of the true mean). The combined standard uncertainty of individual curvature computations is similar for constant-curvature surfaces in 3D: using a 9 ξ-diameter template sphere, u_c for individual computations is ≈ 0.04 ξ^{-1} (±92% and ±124% of the mean curvature value for a 41 ξ-diameter and a 61 ξ-diameter sphere, respectively). Precision error this large can potentially affect the sign of the difference in curvature between two surface sites and, consequently, the sign of the driving force for mass transport between them during sintering processes.

Precision can be increased substantially by using a finer pixel grid. For example, by halving the pixel edge length (from 1 ξ to 0.5 ξ), u_c of individual curvature computations on a 41 ξ-diameter sphere, using a 9 ξ-diameter template, decreases from 0.04 ξ^{-1} to 0.03 ξ^{-1} (or from 92% to 68% of the true curvature). In fact, all the errors described in this section can be reduced by refining the pixel grid. However, such a remedy requires rather large increases in memory allocation. If the pixel edge length is halved, then eight times as many pixels are required to represent the same portion of a 3D microstructure and, furthermore, a scalar machine would spend about eight times as much CPU time to perform the computations over that portion.

One simple way to increase precision consists of, for each pixel P, computing an arithmetic mean of the curvatures computed for P and all the neighboring surface sites within a prescribed radius about P. Figure 4 shows that using an arithmetic mean over progressively larger neighborhoods steadily increases the precision of the curvature computation. The value of u_c for the curvature computations is 0.029 ξ^{-1} (89% of the true curvature) without averaging, and is reduced to 0.003 ξ^{-1} (11% of the true curvature) by averaging over all surface pixels within a radius of 11 ξ. Only slight increases in CPU time are required for averaging. But in complex microstructures, large areas of constant curvature may not exist, so that averaging over large areas may not be feasible without employing a finer pixel grid.

3.2. *Examples*

These various sources of error can all substantially reduce the confidence one may have in the reliability of an *individual* result of this method applied to discretized interfaces. The strength of the method, however, is in accurately reflecting the spatial *distribution* of curvature throughout a complex system composed of one or more interfaces. Where the method has been applied to sintering simulations [3,4] it has provided quite realistic predictions of the direction of mass transport throughout complex 2D and 3D microstructures. Further illustrations of this strength are provided by the following two examples, one in three dimensions and the other in two.

The first example is that of computing the mean curvature of the surface of a digitized sphere. Fig. 5 shows the computed mean curvature of a sphere plotted against its true mean curvature, R^{-1}. The template sphere diameter used was 9 ξ, and Eq. (23) was used to relate the pixel count to the mean curvature (no averaging over neighboring surface pixels was performed). The values shown are averages of the values computed for every surface pixel in one octant, and error bars represent ±1 standard deviation. The average value of mean curvature for each sphere is very accurate, although the precision is low because no averaging was performed over neighboring surface sites. This example is nothing more than a verification of the calibrated relation between pixel count and curvature, Eq. (23). But it does point out that the method should provide a good measurement of, for instance, the average driving force for mass transfer between the various particles comprising a sintering microstructure, although precision error would cause fluctuations about that average value.

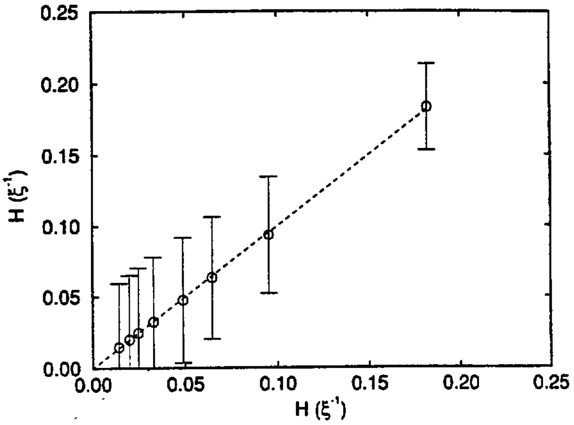

Fig. 5. Computed mean curvature of the surface of a sphere of radius R, plotted against the true mean curvature, $1/R$. Computations were performed using a template sphere with diameter = 9 ξ. Each point represents the arithmetic mean of the mean curvature computed for each pixel along the sphere surface, and error bars indicate ±1 standard deviation.

The second example is that of computing the curvature along a sinusoidal arc in two dimensions, like that analyzed by Mullins [7]. The arc is described by

$$y = f(x) = A\sin(kx). \tag{25}$$

The curvature of an arc described by Eq. (25) can be calculated analytically according to the expression

$$\kappa = \frac{f_{xx}}{(1+f_x^2)^{3/2}} = -\frac{Ak^2\sin(kx)}{(1+A^2k^2\cos^2(kx))^{3/2}}, \tag{26}$$

where f_x and f_{xx} are the first and second derivatives of f with respect to x. Figure 6 shows both the computed and analytical values of the curvature at a number of points along one wavelength $(2\pi k^{-1})$ of the

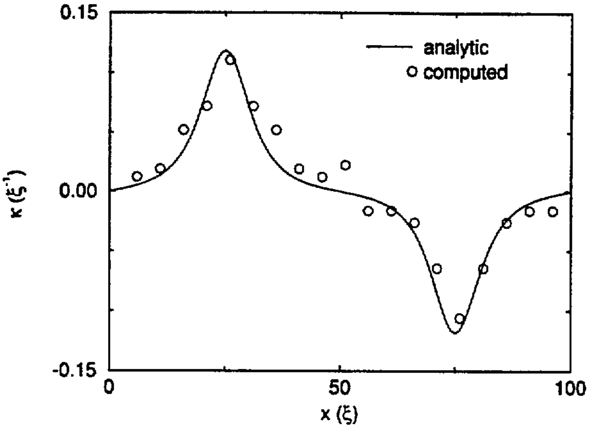

Fig. 6. Computed and analytical values of the curvature for a sinusoidally-perturbed surface ($y = A\sin(kx)$) plotted along one wavelength λ ($2\pi k^{-1}$) of the perturbation. In this graph, $A = 30\,\xi$, $\lambda = 100\,\xi$, and the diameter of the template circle is 15 ξ.

surface. Each individual computed value represents an average over neighboring surface pixels that share either a common edge or corner. Extremal curvature values are predicted quite well by the method (to within 3%), and somewhat greater deviations from the true curvature occur at lower curvature magnitudes.

3.3. *Refinements*

We now describe briefly two refinements to the curvature computation when it is applied to digitized interfaces. The first represents a potential improvement in computational accuracy, while the second allows collection of information concerning the orientation of the interface.

At regions of an interface where sharp fluctuations in curvature occur over small distances, or at regions where two or more separate interfaces are closely spaced, the template method for computing curvature may generate errors not discussed in previous sections. We illustrate the source of this error schematically in Fig. 7. Figure 7(A) shows the region near the point of contact between two spheres. The true mean curvature at point P is simply the curvature of the right-hand sphere, namely B^{-1}. But since the template sphere may overlap a portion of the left-hand sphere, the curvature computation may generate too low a value for the curvature at point P, since it effectively takes the total number of template pixels and subtracts all solid pixels within the template. The curvature computation is especially prone to this error along crack surfaces or in partially-sintered powder compacts. One could, in principle, modestly decrease the error in these regions by decreasing the template radius, but an algorithm that tests for such regions and automatically adjusts the template radius in response to some criterion would likely be computationally intensive.

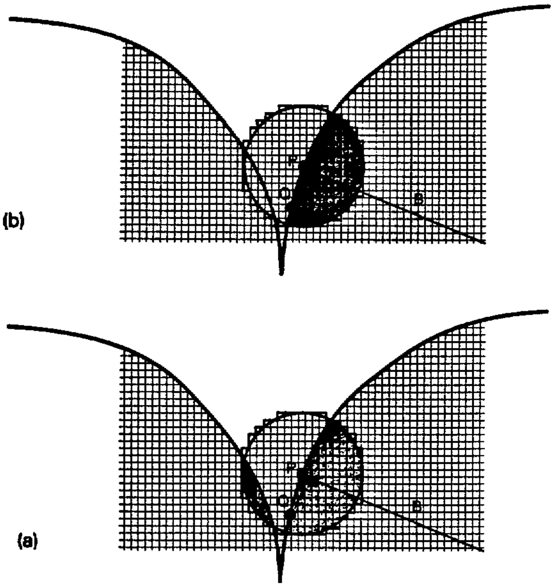

Fig. 7. (A) Position of a template circle used to compute the curvature at point A in the vicinity of the point of contact between two larger circles. The intersection of the template with the left-hand circle will cause an erroneously low computed curvature. (B) Use of a burning algorithm to test for the connectivity of the solid pixels within the template circle. Cross-hatched (burned) pixels are those that are connected to point A. All unburned pixels within the template are counted in the curvature computation.

A simpler way to reduce errors caused by template overlap, without adjusting the template radius, is to incorporate a variant of the “burning” algorithm, frequently used in percolation models to assess the connectivity of a phase in a microstructure. For example, assume that the spheres in Fig. 7 are solid and the region surrounding them is a pore phase composed of some inert vapor. Then applying the burning algorithm at point P, one “burns” all the solid pixels within the template that can be reached from point P without encountering a pore pixel (shown in Fig. 7(B) with darker shading). The difference between the total number of template pixels and the burned solid pixels can then be used to obtain a more accurate estimate of the mean curvature at point P, since the region of overlap with the left-hand sphere does not burn. The burning algorithm can therefore substantially increase the accuracy of the curvature computation near these kinds of microstructural features. For example, in the relatively extreme 2D case of a flat interface of a crack that is one pixel wide, the unmodified method yields a curvature value of -0.442 ξ^{-1}, while including the burning algorithm results in a curvature value of -0.003 ξ^{-1}, which is much closer to the true value of zero for the crack interface. Of course, there are limits to the effectiveness of the burning algorithm. If the template in Fig. 7 were moved several surface pixels closer to the point of contact (point Q), the burning algorithm would not help since all the solid pixels within the template would burn. Template diameter ultimately limits the resolution of the method in these instances.

The burning algorithm can also be applied if all pixels composing a given particle are assigned a unique label. For example, suppose the pixels composing the left particle in Figure 7 are assigned a different label than those composing the right particle, indicating it is a separate grain. By burning only pixels with like labels, the burning algorithm applied at Q could again produce a more accurate curvature value than that produced by the unmodified method. Assigning different labels to pixels in different particles is particularly useful in sintering and grain growth simulations where grain boundaries can be identified as the interface between two solid regions with different pixel labels [24,25]. The curvature along the grain boundaries can then also be computed using the burning algorithm.

Finally, the orientation of an interface can be both an important thermodynamic and kinetic factor when the specific interfacial free energy, γ, is anisotropic. In such cases theory predicts [26], and experiments confirm [27,28], that interface orientations with lower values of γ are favored over those of high γ. It is straightforward to compute the orientation, relative to some fixed coordinate frame, of the surface normal vector, $\hat{\mathbf{n}}$, erected at any point P along a digitized surface. To determine the orientation of $\hat{\mathbf{n}}$, one can simply keep track of the x, y, and z positions, relative to P, of each pixel in the template volume as it is counted. Using P as the origin, the direction of $\hat{\mathbf{n}}$ is given by the 3-tuple $(X|\hat{\mathbf{n}}|^{-1}, Y|\hat{\mathbf{n}}|^{-1}, Z|\hat{\mathbf{n}}|^{-1})$, where $|\hat{\mathbf{n}}| = (X^2+Y^2+Z^2)^{1/2}$ and, for example,

$$X = \sum_{i=1}^{m} x_i \tag{27}$$

and m is the number of pixels counted in the curvature computation at P. The orientation of $\hat{\mathbf{n}}$ is then specified by, for instance, the two angles

$$\phi = \tan^{-1}\left(\frac{Y}{X}\right),$$
$$\theta = \tan^{-1}\left(\frac{\sqrt{X^2+Y^2}}{Z}\right) \tag{28}$$

used in a spherical polar coordinate frame (see Fig. 8(A)). Figure 8(B) shows the computed orientation ϕ of $\hat{\mathbf{n}}$ along the perimeter of a circle in 2D. To construct this figure, the diameter of the circle was chosen as 121 ξ, and the template circle diameter was 13 ξ. The computed values of ϕ are correct to within ±3%. Figure 8(C) shows the computed value of ϕ for two different values of θ, and again the computed values of ϕ are quite accurate. Because it uses much of the same data and is accomplished at the same time as the curvature computation, determination of the surface normal orientation requires relatively little additional computational effort.

4. Summary

A conceptually and computationally simple numerical technique for computing mean interfacial curvature has been demonstrated. The procedure requires no detailed information about the shape of the interface, and consists of determining the portion of volume enclosed by a template sphere (or circle in 2D) that lies on one side of the interface. The mean curvature of the interface, relative to the phase lying on the opposite side, is then approximately linear in this computed volume. Higher-order correction terms to the linear approximation have been shown to be negligibly small in most cases for appropriate choice of the template radius. Application of the procedure to digitized representations of interface images are straightforward and yield accurate estimates of the spatial distribution of curvature along interfaces, although errors in individual measurements arise from the discrete representation of the interface. These errors can all be reduced by refining the pixel grid, at the expense of increased memory allocation and CPU time. Incorporation of a burning algorithm has been shown to provide greater accuracy in regions having sharp curvature fluctuations, like those along sharp crack-like features. A simple extension of the algorithm allows accurate computation of the interface *orientation* relative to a fixed reference frame.

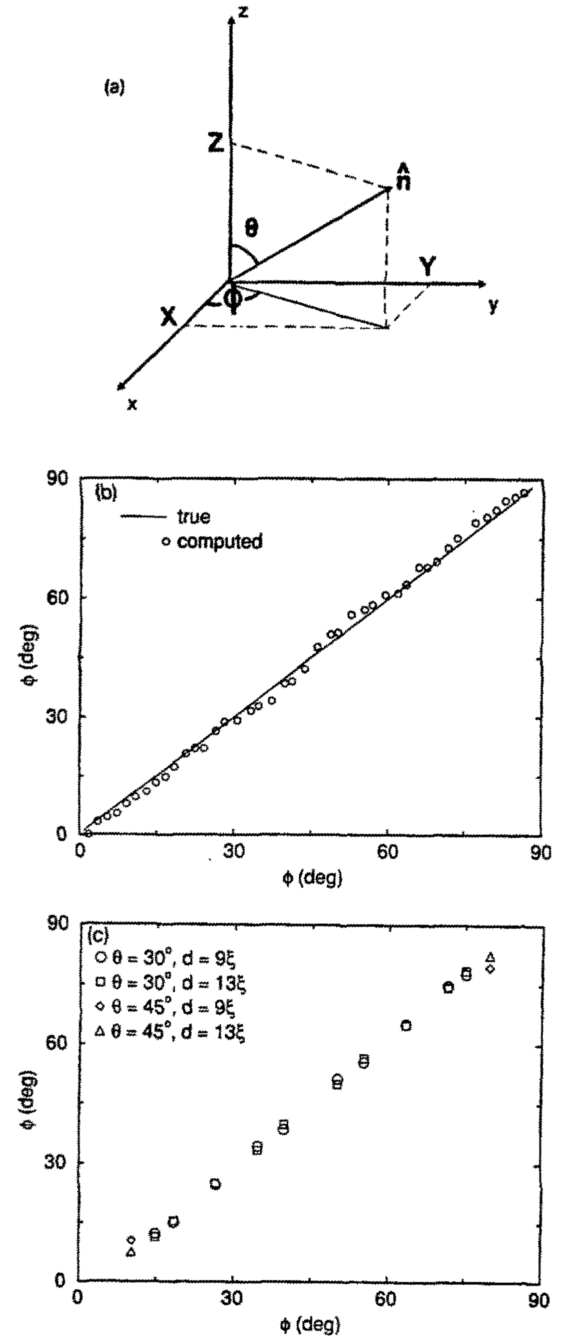

Fig. 8. (A) Angles θ and ϕ used to describe the orientation of surface normal vector $\hat{n}$. (B) Computed angle ϕ of the surface normal at a point P along the perimeter of a circle in the xy-plane, relative to the x-axis, plotted vs. the true value of ϕ given by $\tan^{-1}(Y/X)$, where X and Y are the $x-$ and y-position of the surface pixel relative to the circle center. The template circle diameter $d = 15\ \xi$. (C) Values of ϕ computed at two different values of $\theta = 30^o$ and $\theta = 40^o$ on a sphere with diameter 61 ξ . Template sphere diameters are $d = 9\ \xi$ and 13 ξ.

Acknowledgements

JWB would like to thank the National Research Council for postdoctoral fellowship support. The authors also thank Tim Burns for his contributions to the error analysis, and Andy Roosen for a thoughtful critique of the manuscript.

References

[1] E.J. Garboczi and D.P. Bentz, Mater. Res. Soc. Bull. 18(3) (1993) 50.

[2] D.P. Bentz and E.J. Garboczi, Cement and Concrete Res. 21 (1991) 325.

[3] D.P. Bentz, P.J.P. Pimienta, E.J. Garboczi and W.C. Carter, Proc. Mater. Res. Soc. 249 (1991) 413.

[4] P.J.P. Pimienta, E.J. Garboczi and W.C. Carter, Comput. Mater. Sci. 1 (1992) 63.

[5] J.E. Taylor, Acta Metall. Mater. 40 (1992) 1475.

[6] C. Herring, in: The Physics of Powder Metallurgy, ed. W.E. Kingston (McGraw-Hill, New York, 1951) pp. 143–179.

[7] W.W. Mullins, in: Metal Surfaces: Structure, Energetics and Kinetics (ASM, Metals Park, Ohio, 1963) pp. 17–64.

[8] W.D. Kingery, H.K. Bowen and D.R. Uhlmann, in: Introduction to Ceramics, 2nd Ed. (Wiley-Interscience, New York, 1976) p. 452.

[9] I.M. Lifshitz and V.V. Slyozov, J. Phys. Chem. Solids 19 (1961) 35.

[10] G. Lewis and M. Randall, in: Thermodynamics 2nd Edition, Ch. 29, Revised by K.S. Pitzer and L. Brewer (McGraw-Hill, New York, 1961).

[11] W.D. Kingery and M. Berg, J. Appl. Phys. 26 (1955) 1205.

[12] W.C. Carter and R.M. Cannon, in: Ceramic Transactions, Sintering of Advanced Ceramics, Vo. 7, eds. C.A. *Handwerker, J. Blendell and W. Kayser (Am. Ceram. Soc.,* Westerville, Ohio, 1990) pp. 137–163.

[13] B.J. Kellett and F.F. Lange, J. Am. Ceram. Soc. 72 (1989) 725.

[14] R. Tao, M.A. Novotny and K. Kaski, Phys. Rev. A 38 (1988) 1019.

[15] D. Bouvard and R.M. McMeeking, "The Deformation of Interparticle Necks by Diffusion Controlled Creep," unpublished work.

[16] L.P. Kadanoff, J. Stat. Phys. 39 (1985) 267.

[17] T. Vicsek, Phys. Rev. Lett. 53 (1985) 2281.

[18] T. Vicsek, Phys. Rev. A 32 (1985) 3084.

[19] E.A. Holm, D.J. Srolovitz and J.W. Cahn, Acta Metall. Mater. 41 (1993) 1119.

[20] C.-T. Wang, in: Applied Elasticity, Ch. 12 (McGraw-Hill, New York, 1953).

[21] A. Onuki, Phys. Rev. A 45 (1992) 16.

[22] B.N. Taylor and C.E. Kuyatt, Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results, NIST Technical Note 1297 (NIST, Gaithersburg, Maryland, 1993).

[23] D. Stauffer, in: Introduction to Percolation Theory (Taylor and Francis, London, 1985) pp. 3–11.

[24] M.P. Anderson, D.J. Srolovitz, G.S. Grest and P.S. Sahni, Acta Metall. 32 (1984) 783.

[25] G.N. Hassold, I.-W. Chen and D.J. Srolovitz, J. Am. Ceram. Soc. 73 (1990) 2857.

[26] G. Wulff, Z. Kristallogr. 34 (1901) 449.

[27] B.E. Sundquist, Acta Metall. 12 (1964) 67.

[28] J.C. Heyraud and J.J. Metois, Acta Metall. 28 (1980) 1789.